

Contents

Contents	1
1. Scope.....	1
2. References	1
3. Terms and Definitions	1
4. Abbreviations.....	1
5. Introduction.....	2
6. Conversion CAP to AEA.....	2
7. OTT Emergency Alert Web Server TV 3.0.....	6
8. Broadcaster Station Operation	7
8.1. OTA Operation.....	7
8.2. OTT Operation.....	8
9. Receiver Operation	8
9.1. Alert Monitoring.....	8
9.1.1. Wake-up field.....	10
9.2. Presentation Model.....	10
9.2.1. Minor Priority.....	11
9.2.2. Low Priority	12
9.2.3. Moderate Priority	13
9.2.4. High Priority	17
9.2.5. Maximum Priority	19
Annex A - Regulatory Compliance Requirements for Brazil	22
A.1 AEAT configuration and format in Brazil.....	23
A.2 Broadcaster Station Operation.....	23
Annex B - Conversion script from CAP to AEAT.....	24

1. Scope

This document is a companion to ABNT NBR 25607:2025 and its annexes, providing complementary implementation guidelines intended to clarify architectural intent and operational behavior. It does not define normative requirements, but describes recommended practices, supports consistent interpretation of the standard, and promotes interoperable TV 3.0 Application-oriented Platform (AoP) implementations.

2. References

The following documents are cited in the text in such a way that their contents, in whole or in part, constitute requirements for this document. For dated references, only the editions cited apply. For undated references, the most recent editions of that document (including amendments) apply.

ABNT NBR 25607:2025, *TV 3.0 – Emergency Warning System*

ABNT NBR 25608:2025, TV 3.0 – Application Coding

ABNT NBR 25609:2025, TV 3.0 – Receivers

ATSC A/331:2025-10, Signaling, Delivery, Synchronization, and Error Protection

ATSC A/351:2025-07, Techniques for Signaling, Delivery, and Synchronization

ATSC 3.0 Advanced Emergency Information System Implementation Team Guide, Doc. AEA-IT-024r31, 20 February 2019.

Recommendation ITU-T X.1303 bis, Common alerting protocol (CAP 1.2)

SBTVD TV3.0 OG – 06:2025 - TV 3.0 Operational Guidelines - Closed Signing

3. Terms and Definitions

For the purposes of this Document, the terms and definitions described at ABNT NBR 25607 should be used.

Other terms are described as follows:

OTT Alert Web Server TV 3.0. The broadcaster's OTT alert web server. This server is used to provide receivers with a URL for monitoring OTT alerts. The URL for this server is sent in the `@url` attribute of the `emergencyWarningTransmission` element in the BAM.

Emergency Alert Authorities Web Server. Authority agencies responsible for providing initial information on emergency alerts. They should be consulted by broadcasters, not by receivers. They may operate at the national or local level.

4. Abbreviations

For the purposes of this Document, the following abbreviations apply.

AEA	Advanced Emergency informAtion
AEAT	Advanced Emergency informAtion Table
ATSC	Advanced Television Systems Committee
bzs	Brazilian Sign Language
CAP	Common Alert Protocol
COBRADE	Brazilian Disaster Classification and Coding
DTV	Digital TeleVision
OTA	Over the Air
OTT	Over the Top
URL	Uniform Resource Locator

5. Introduction

This document is a companion to ABNT NBR 25607:2025 and its annexes, providing complementary implementation guidelines intended to clarify architectural intent and operational behavior. It does not define normative requirements, but describes recommended practices, supports consistent interpretation of the standard, and promotes interoperable TV 3.0 platform implementations.

The operational guidelines addressing the architectural aspects of TV 3.0 Emergency Warning System are presented in the main body of this document. Guidelines related to specific technologies and regulatory aspects employed in TV 3.0 Emergency Warning System are provided in the annexes that follow.

- Annex A contains the guidelines on Regulatory Compliance Requirements for Brazil.
- Annex B contains a Conversion script from CAP to AEAT.

6. Conversion CAP to AEA

The equivalence conversion table between CAP (Common Alert Protocol) and AEA is presented in Table 1. The specifications following the table provide the semantics of the fields.

Table 1 - Equivalence table between CAP and AEA

CAP fields	AEA fields	Observations
No equivalence	AEAT@aeaTableId	
Message ID (cap.alert.identifier.identifier)	AEAT.AEA@aeaId	
Sender ID (cap.alert.sender.identifier)	AEAT.AEA@sender	
Source (cap.alert.source.identifier)	AEAT.AEA@issuer	
Effective Time (cap.alertInfo.effective.time) (priority 1)	AEAT.AEA. Header@effective	Fields with priority indicated must apply equivalence using the lowest priority. When the field with the lowest priority is not present in the CAP, the next priority field must be used.
Onset Time (cap.alertInfo.onset.time) (priority 2)		
Sent Date (cap.alert.sent.time) (priority 3)		
Expiration Time (cap.alertInfo.expires.time)	AEAT.AEA.Header@expires	
Message Status (cap.alert.status.code)	No equivalence	
Message Type (cap.alert.msgType.code)	AEAT.AEA@aeaType	

Scope (cap.alert.scope.code)	AEAT.AEA@audience	
Restriction (cap.alert.restriction.text) (concat)	AEAT.AEA@subAudience	Fields with the “concat” label, when presented, should be concatenated following the order presented in this table, to create the equivalence. The strings should be separated by the space character.
Addresses (cap.alert.addresses.group) (concat)		
Audience (cap.alertInfo.audience.text) (concat)		
Handling Code (cap.alert.code.code)	No equivalence	
Note (cap.alert.note.text)	No equivalence	
Reference IDs (cap.alert.references.group)	AEAT.AEA@refAEAIId	
Incident IDs (cap.alert.incidents.group)	No equivalence	
Event Category (cap.alertInfo.category.code)	AEAT.AEA@category	
No equivalence	AEAT.AEA@wakeUp	True, if AEAT.AEA@priority is maximum. False otherwise.
Language (cap.alertInfo.language.code)	AEAT.AEA.AEAText@lang	
	AEAT.AEA.Header.EventDesc@lang	
Event Type (cap.alertInfo.event.text)	AEAT.AEA.Header.EventDesc	
Event Code (cap.alertInfo.eventCode.code.valueName)	AEAT.AEA.Header.EventCode@type	
Event Code (cap.alertInfo.eventCode.code.value)	AEAT.AEA.Header.EventCode	
Severity (cap.alertInfo.severity.code)	AEAT.AEA@priority	See Table 2.
Urgency (cap.alertInfo.urgency.code)	Partially used for AEAT.AEA@priority	See Table 2.

Certainty (cap.alertInfo.certainty.code)	No equivalence	
Headline (cap.alertInfo.headline.text) (concat)	AEAT.AEA.AEAText	Fields with the “concat” label, when presented, should be concatenated following the order presented in this table, to create the equivalence. The strings shall be concatenated separated by the space character.
Event Description (cap.alertInfo.description.text) (concat)		
Instructions (cap.alertInfo.instruction.text) (concat)		
No equivalence	AEAT.AEA.LiveMedia	
Resource (cap.alertInfoResource.resource.group)	AEAT.AEA.Media	
Language (cap.alertInfo.language.code)	AEAT.AEA.Media@lang	
Description (cap.alertInfoResource.resourceDesc.text)	AEAT.AEA. Media@mediaDesc	
No equivalence	AEAT.AEA. Media@mediaType	
Mime Type (cap.alertInfoResource.mimeType.identifier)	AEAT.AEA. Media@contentType	
URI (cap.alertInfoResource.uri.identifier)	AEAT.AEA.Media@url	
File Size (cap.alertInfoResource.size.integer)	AEAT.AEA. Media@contentLength	
No equivalence	AEAT.AEA. Media@mediaAssoc	
No equivalence	AEAT.AEA. Media@alternateUrl	
Dereferenced URI (cap.alertInfoResource.derefUri.data)	No equivalence	
Digest (cap.alertInfoResource.digest.code)	No equivalence	

Area (cap.alertInfoArea.area.group)	AEAT.AEA.Header.Location	
Area Description (cap.alertInfoArea.areaDesc.text)	No equivalence	
Area Polygon (cap.alertInfoArea.polygon.group)	AEAT.AEA.Header.Location and AEAT.AEA.Header.Location@type	
Area Circle (cap.alertInfoArea.circle.group)		
Area Geocode (cap.alertInfoArea.geocode.code)		
Altitude (cap.alertInfoArea.altitude.quantity)	No equivalence	
Ceiling (cap.alertInfoArea.ceiling.quantity)	No equivalence	

The value of the CAP Language field (cap.alertInfo.language.code) must be set in three AEAT fields, AEAT.AEA.AEAText@lang, AEAT.AEA.Header.EventDesc@lang and AEAT.AEA.Media@lang.

The CAP Severity field (cap.alertInfo.severity.code) should have its values related to the possible AEAT priority values (AEAT.AEA@priority) according to the equivalence described in Table 2. As specified in ATSC A/331:2025-10, Table 6.13, for AEAT.AEA@priority, the Maximum and High priorities also depends on the urgency being immediate.

Table 2 - Equivalence between the values of the fields cap.alertInfo.severity.code and AEAT.AEA@priority

Cap.alertInfo.severity.code (string)	AEAT.AEA@priority (interger)
“Extreme” and cap.alertInfo.urgency.code equals to Immediate.	4 (Maximum)
“Severe” and cap.alertInfo.urgency.code equals to Immediate.	3 (High)
“Moderate”	2 (Moderate)
“Minor”	1 (Low)
“Unknown”	0 (Minor)

Each field Area Polygon (cap.alertInfoArea.polygon.group), Area Circle (cap.alertInfoArea.circle.group), Area Geocode (cap.alertInfoArea.geocode.code) should be

converted to a pair of fields of `AEAT.AEA.Header.Location` and `AEAT.AEA.Header.Location@type`. Multiples `AEAT.AEA.Header.Location` and `AEAT.AEA.Header.Location@type` can be used.

An example of conversion script has been made available in Annex B of this Document. It takes an XML file in CAP format as input and generates an XML file in AEAT format as output, containing the conversion results.

7. OTT Emergency Alert Web Server TV 3.0

This server is used to provide receivers with a URL for monitoring OTT alerts. The URL for this server is sent in the `@url` attribute of the `emergencyWarningTransmission` element in the BAM.

It is recommended that the OTT Emergency Alert Web Server TV 3.0 provide an HTTP URL in the HTTP GET format.

The HTTP route should have the following format:

```
https://<host>/<state>/<city>
```

Where:

- **state** should be the two-letter state abbreviation.
- **city**, the name of the city without spaces or special characters. The first letter of each word should be capitalized.

Examples:

- `https://alertastv3.com.br/SP/Campinas`
- `https://alertastv3.com.br/PB/JoaoPessoa`

It is recommended that, if the `<city>` path parameter is missing, a file containing alerts for all cities in the state within the broadcaster's coverage area be returned. And, if both the `<state>` and `<city>` path parameters are missing, a file containing all alerts for the broadcaster's coverage area at the national level should be returned.

It is recommended that the route not queries the database, but instead returns the content of the data stored as a file. Thus, the server must have the files for each city and each state ready.

According to ABNT NBR 25607, section A.2.3.1, the returned body should be an XML in AEA or CAP format.

It is recommended that the HTTP response header contains the `RefreshTime` parameter, indicating the interval, in seconds, after which the receiver should repeat the query to check for new alerts. This parameter is controlled by the server; its value can be adjusted according to demand, aiding in load balancing. The recommended initial default value is 30 seconds.

8. Broadcaster Station Operation

Broadcasters transmitting emergency alerts via OTA or OTT are advised to follow the following operating protocol.

1. Monitor the Emergency Alert Authorities Web Server.
2. Determine whether the alert should be broadcast by the station, taking the following points:

- a. The location of the alert must be within the broadcaster coverage area. To this end, it is recommended that the broadcaster verify whether the polygon defining the region affected by the alert intersects with the broadcaster's coverage area.
3. If the alert is not in AEAT format, it must be converted to AEAT. Section 6 provides a conversion table from CAP to AEAT.
4. Add sign language accessibility to the alert using one of the following formats:
 - a. Add a gloss language to the `AEA.AEAText` and `AEA.Header.EventDesc` fields. To do this, the `AEA.AEAText@lang` and `AEA.Header.EventDesc@lang` should be configured with the value indicating the sign language, for example, "bzs", which indicates Brazilian Sign Language. The Sign Language (Closed Signing) Operational Guide contains recommendations for the process of translating from Portuguese to gloss sign language.
 - b. Submission of a video file containing a human interpreter or a digital avatar interpreting sign language. To do this, the `AEA.Media@url` field must be set to the video's URL, and the `AEA.Media@lang` field must be set to the sign language, for example, "bzs".
5. Broadcasters can add interactive content and new media to enhance the alert.

Depending on the type of broadcast the broadcaster is conducting — OTA or OTT — it is recommended to follow the protocol outlined in the following sections.

8.1. OTA Operation

To transmit the alert via OTA, it is recommended that you follow the operating protocol below.

1. The alert in AEA format must be signaled by the broadcaster, performing the necessary adjustments:
 - a. LLS table. The AEAT should be inserted into the LLS table.
 - b. NRT Files (media). The media included in the alert can be transmitted as NRT Files.
 - c. Wake-up field. Composed by the 2 wake-bits (`ea_wake_up_1` and `ea_wake_up_2`) of the bootstrap physical layer. Should be configured as:
 - i. 00 – No active emergency message
 - ii. 01, 10, and 11 – Rotating through these values will inform the receiver that there is either a new emergency message or that there is new and substantial information being added to an existing message, as specified in ATSC 3.0 Advanced Emergency Information System Implementation Guide (AEA-IT-024r31), Section 7.2.
2. Alert transmission rate (AEAT):
 - a. Alerts with wake-up true: 1 second during the first 2 minutes of the wake-up field change. After this time, the AEAT transmission rate must be set to 10 seconds, as specified in ATSC A/331:2025-10, Section G.2.
 - b. Alerts with wake-up false: 5 seconds, as recommended in Section 4.10 of ATSC A/351.
 - c. The alert does not need to be transmitted after its expiration time has been reached.

8.2. OTT Operation

To broadcast the alert via OTT, we recommend following this operating protocol.

1. The broadcaster provides its own static OTT Alert Web Server TV 3.0, in accordance with the recommendations in Section 8. The broadcaster may have a single server at the national level or multiple servers distributed among its affiliate stations to decentralize requests.
2. The `@url` attribute of the BAM's `emergencyWarningTransmission` element is set to the web address of the broadcaster's OTT Alert Web Server TV 3.0. It is recommended that the URL already includes the state and city information configured according to the geographical region.

3. The broadcaster's OTT Alert Web Server TV 3.0 should periodically monitor the Emergency Alert Authorities Web Server, every 30 seconds and generate or update the files for the cities and states within the station's coverage area. If the same alert applies to more than one city, it must be included in each city's file.

9. Receiver Operation

It is recommended that the receiver follow the guidelines below for monitoring and displaying emergency alerts.

9.1. Alert Monitoring

It is recommended that receivers defined in ABNT NBR 25609 follow the guidelines for monitoring emergency alerts, in accordance with Table 3.

Table 3 - Recommendations for monitoring emergency alerts.

Condition	Number of available tuners	Monitoring OTA transmission	Monitoring OTT transmission
Receiver on standby	0	Not applicable	Perform monitoring via OTT from any of the Bootstrap Applications that provide a monitoring URL, at the manufacturer's discretion.
Receiver on standby	1 or more	With any of the available tuners, tune to the physical channel with the strongest signal among those that have DTV service that broadcasts emergency alerts. Monitor the Wake-up Field of the Bootstrap Signal [REFA331].	Not applicable
Receiver turned on and outside the context of DTV service.	0	Not applicable	Perform monitoring via OTT from any of the Bootstrap Applications that provide a monitoring URL, at the manufacturer's discretion.
Receiver turned on and outside the context of DTV service.	1 or more	With any of the available tuners, tune to the physical channel with the strongest signal among those that carry DTV service, which broadcasts emergency alerts. Monitor alerts on this service.	Not applicable
Receiver turned on and in the context of a DTV service, that indicates it is an OTA emergency alert transmitter.	0	Monitor alerts on the current service via OTA.	Not applicable
Receiver turned on and	0	Not applicable	Perform monitoring via OTT from

in the context of a DTV service, that does not indicate it is an OTA emergency alert transmitter.			any of the Bootstrap Applications that provide a monitoring URL, at the manufacturer's discretion.
Receiver turned on and in the context of a DTV service, that indicates it was an OTA emergency alert transmitter.	1 or more	Monitor alerts on the current service via OTA.	Not applicable
Receiver turned on and in the context of a DTV service, that does not indicate it is an OTA emergency alert transmitter.	1 or more	With any of the available tuners, tune to the physical channel with the strongest signal among those that carry DTV service, which broadcasts emergency alerts. Monitor the alerts on this service.	Not applicable

When performing OTT monitoring, it is recommended to access the alert via the `@url` attribute of the `emergencyWarningTransmission` element in the BAM, as defined in ABNT NBR 25608. The monitoring interval can be set according to the `RefreshTime` parameter in the HTTP header, as indicated in Section 8.

The `@url` specified in the BAM is expected to already include city and/or state information, so the receiver does not need to add any path parameters to the URL.

Upon receiving an alert, it is recommended to verify whether the alert should be displayed by the receiver, considering the following points:

1. The receiver's geographic region overlaps with the alert region.
2. The alert audience (`AEA@audience`) is compatible with the recipient's audience.

9.1.1. Wake-up field

The wake-up field, composed of the 2 wake-up bits (`ea_wake_up_1` and `ea_wake_up_2`) of the bootstrap physical layer, should be monitored by the receiver even in the stand-by mode. In this case, monitoring should be performed at two minute intervals, as recommended in Section 9.2.2 of ATSC 3.0 Advanced Emergency Information System Implementation Guide (AEA-IT-024r31). This period is a tradeoff between energy consumption in the receiver and timeliness of detecting emergency messages.

Upon receiving the wake-up signal, the receiver should ignore both the alert and the wake-up signal if the alert does not satisfy conditions 1 and 2 specified in section 9.1. In this case, the wake-up signal should only be considered again if there is a change in the pattern of the wake-up bits.

9.2. Presentation Model

The recommendations for the visual presentation of alerts, for each AEAT priority level (`AEAT.AEA@priority`), are outlined in the following sections.

It is recommended that media (images and videos), deep-linked apps, and sign language be used for alerts with Moderate, High, and Maximum priority levels.

The main textual information that should be displayed in alerts consists of the following attributes:

1. `AEAT.AEA.Header.EventDesc`
2. `AEAT.AEA.AEAText`

These attributes should be displayed in a window on the television screen according to their priority level and in accordance with the display guidelines presented in this section.

In accordance with ABNT NBR 25607, these attributes should be synthesized by voice when the alert priority is maximum, or when the audio description feature is active for at least one viewer profile, or when the receiver's text-to-speech feature is active.

If the receiver supports sign language display, a sign language window should be displayed alongside the alert, provided that the priority level is moderate, high, or maximum. Examples of this display are presented in the following sections.

If the alert includes video or image media, these should be displayed.

If a deep link to an application is provided in the `AEAT.AEA.Media@url` attribute, the referenced application should be accessible via a button.

9.2.1. Minor Priority

The display recommendation for low-priority alerts is shown in Figure 1.

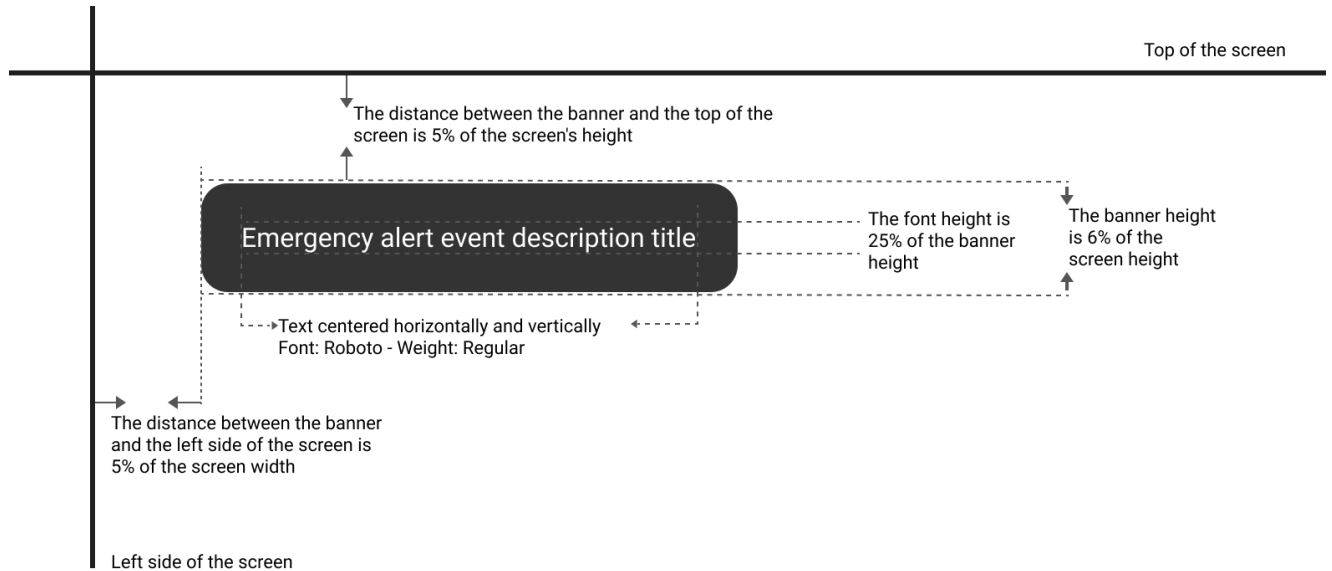


Figure 1 - Recommended positioning of the alert window, font, and screen aspect ratio.

An example of the TV alert is shown in Figure 2.



Figure 2 - Display of a minor priority alert on a 16:9 TV.

9.2.2. Low Priority

The display recommendation for low-priority alerts is shown in Figure 3.

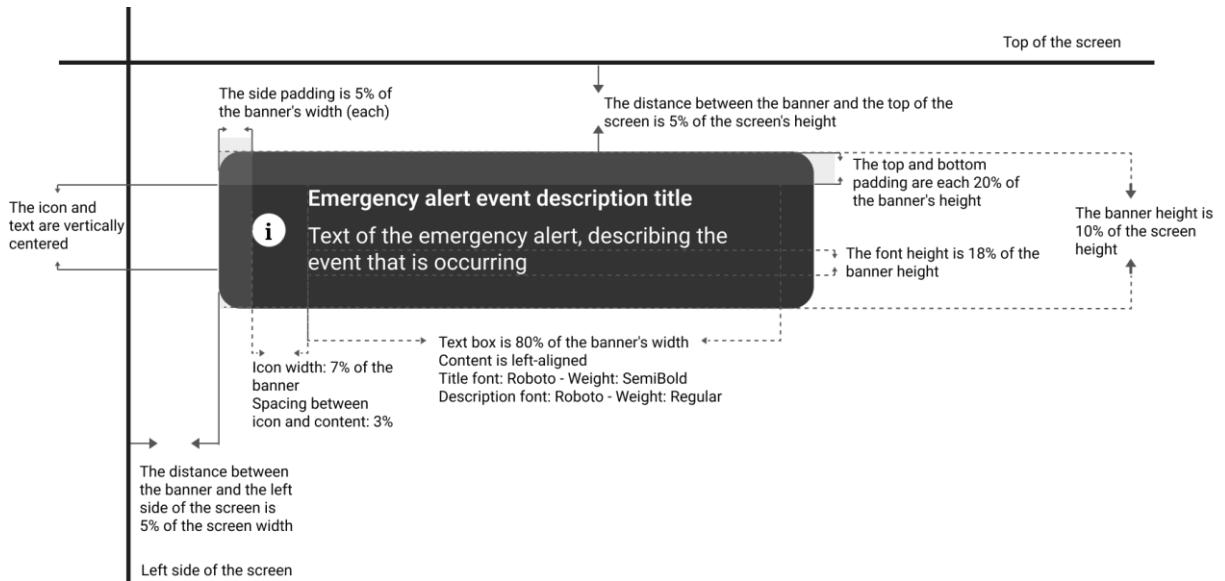


Figure 3 – Recommended positioning of the alert window, font and screen aspect ratio.

An example of the alert on TV is shown in Figure 4.

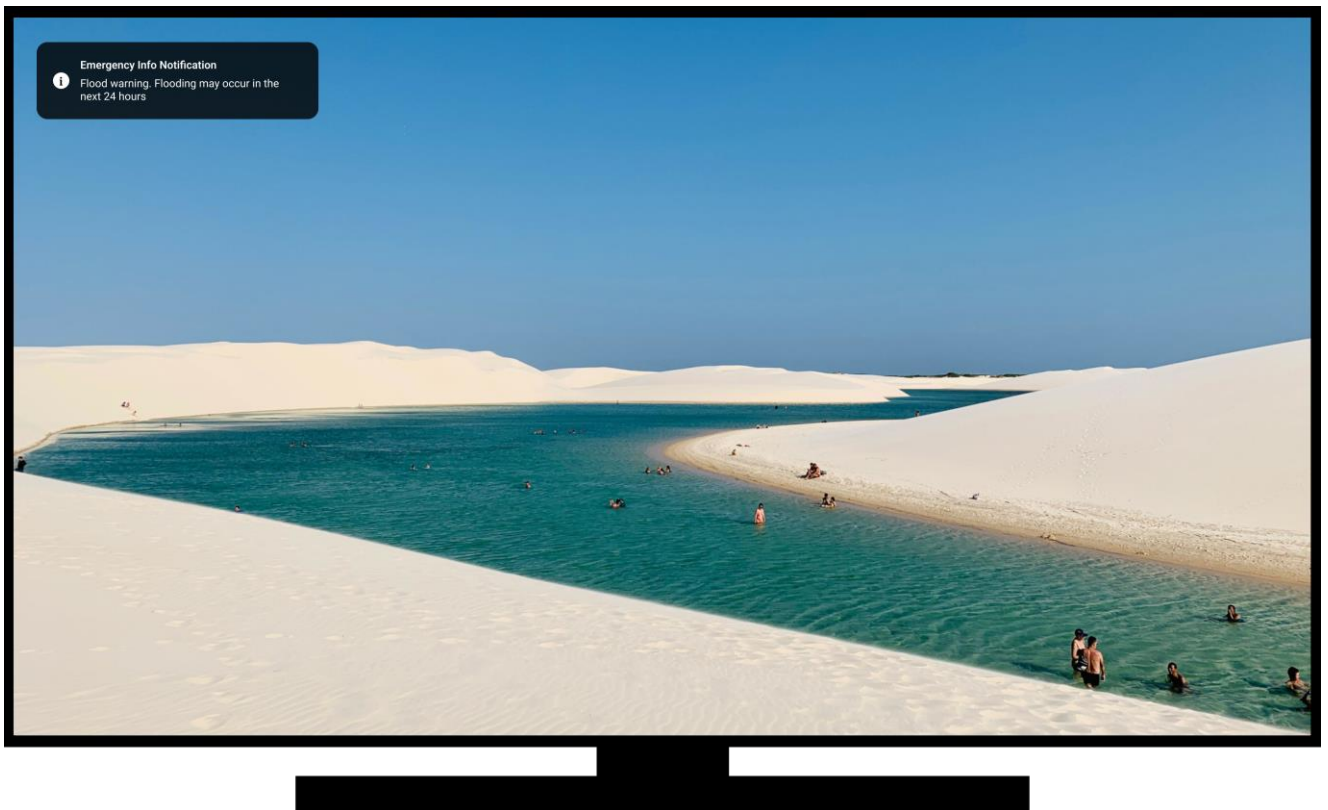


Figure 4 - Display of a low-priority alert on a 16:9 TV.

9.2.3. Moderate Priority

The display recommendations for moderate-priority alerts are shown in Figures 5 to 11. Figure 5 shows the proposed standard layout, including recommendations for positioning, font, aspect ratio, and the 'Details' and 'Close' options.

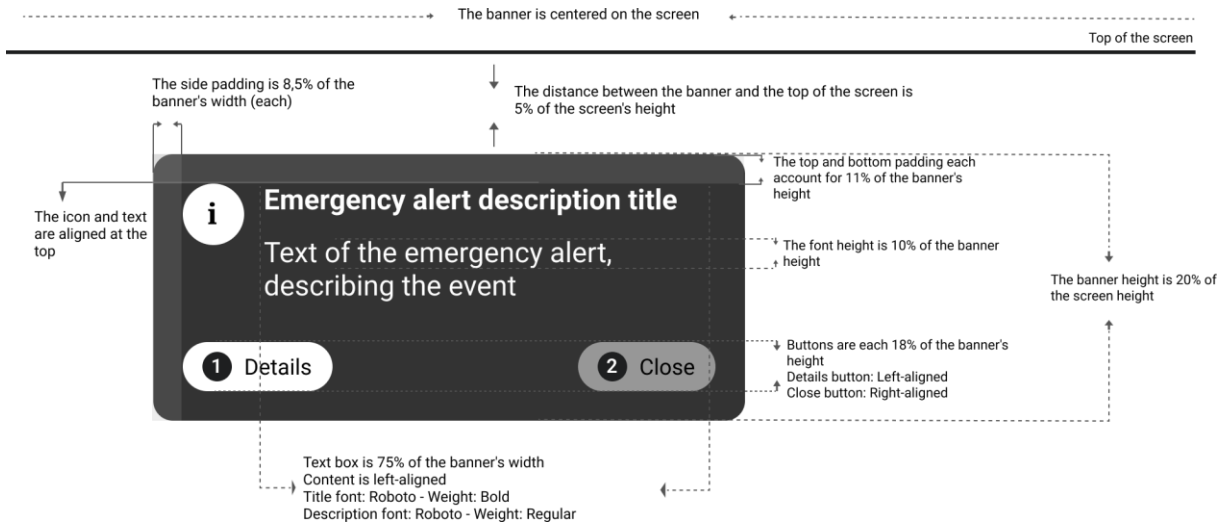


Figure 5 – Recommended positioning of the alert window, font and screen aspect ratio.

Figure 6 shows the alert with the 'Details' mode enabled; it can be seen that more of the alert text is displayed.

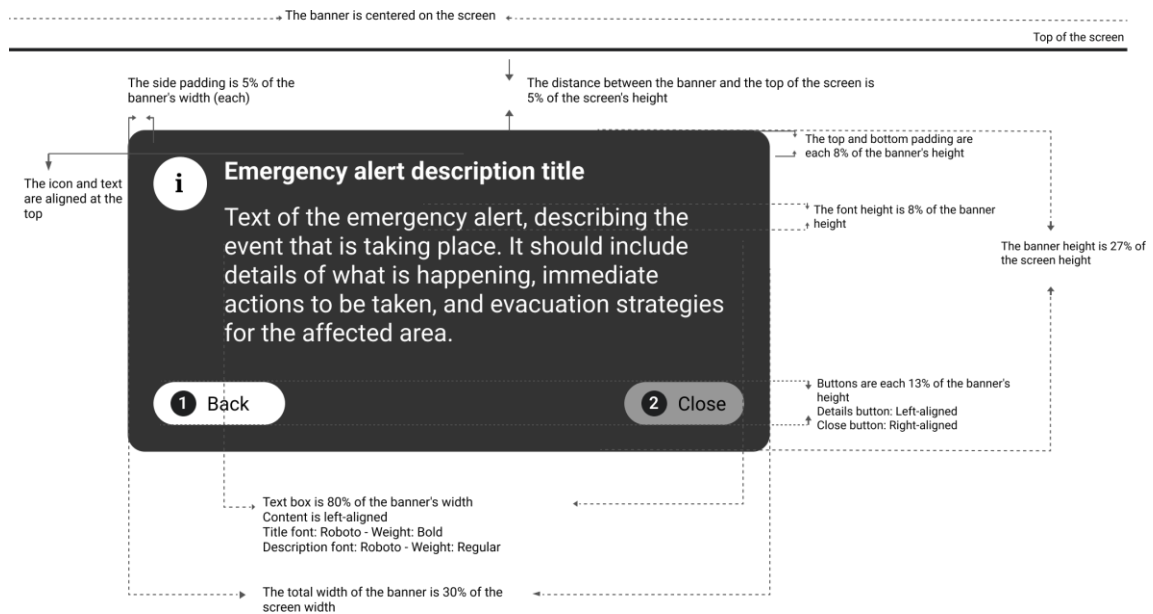


Figure 6 – Recommended positioning of the alert window in detail mode.

It is recommended that the sign language window be displayed if any of the receiver profiles have sign language accessibility enabled. Figure 7 illustrates this scenario.

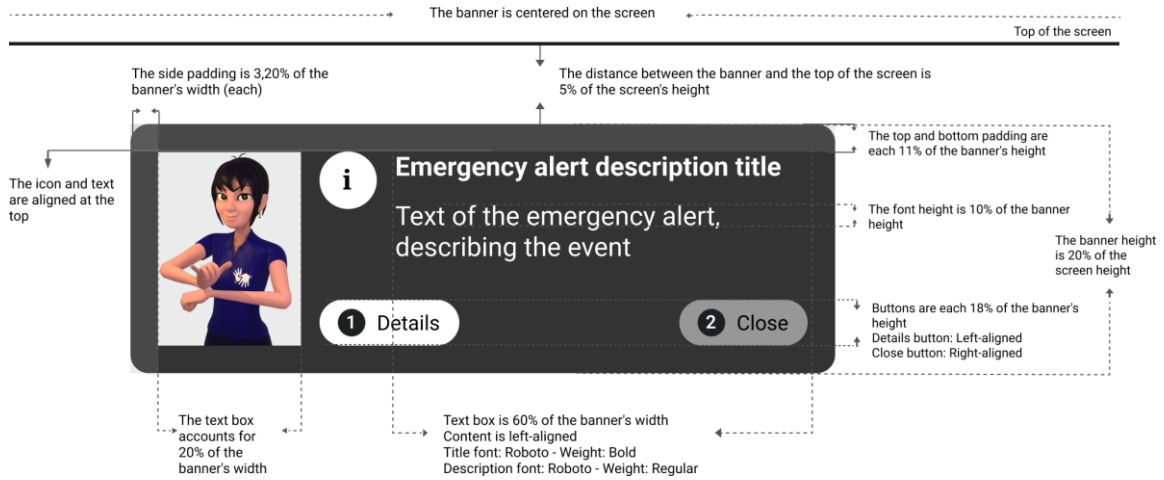


Figure 7 – Recommended positioning of the alert window, font and screen aspect ratio with the sign language window.

Figure 8 shows the alert window with sign language and the 'Details' mode enabled.

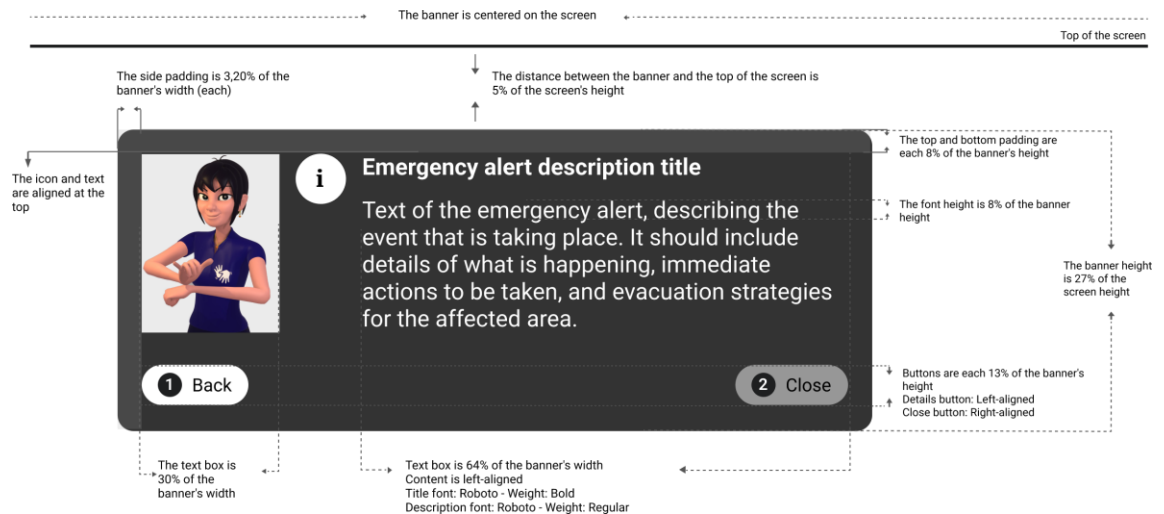


Figure 8 – Recommended positioning of the alert window in detail mode, with a sign language window.

Where the alert contains images or videos, these can be displayed via an image gallery, as shown in Figure 9.

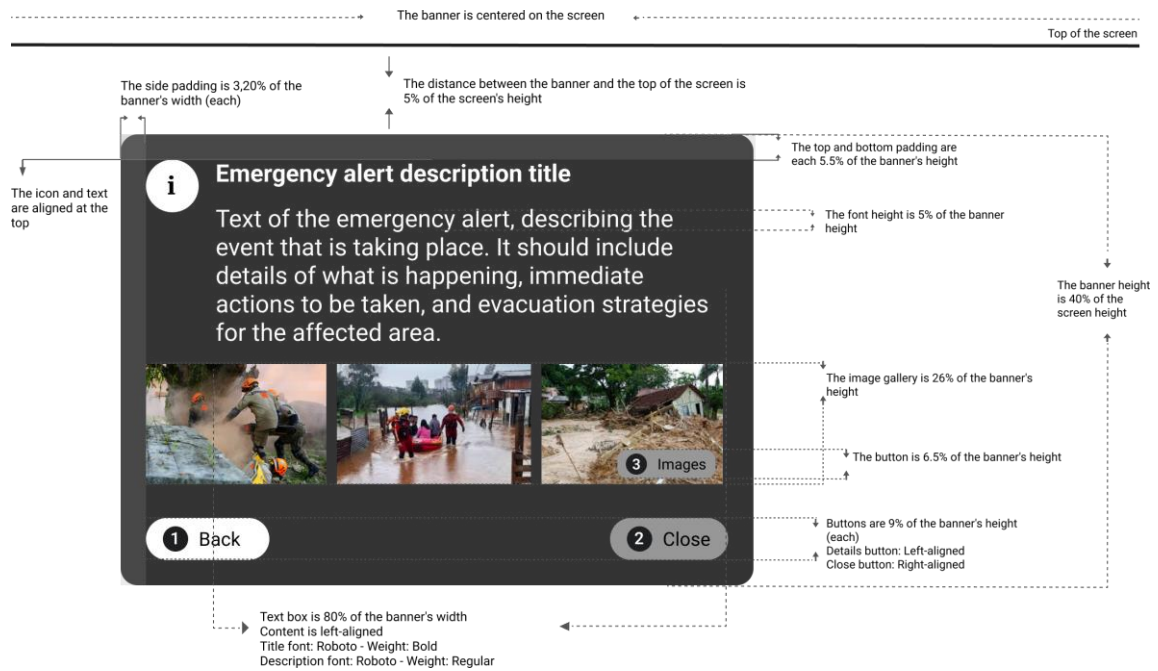


Figure 9 – Recommended positioning of the alert window when images are displayed.

If the alert contains deep links to applications, these can be indicated using buttons, as shown in Figure 10.

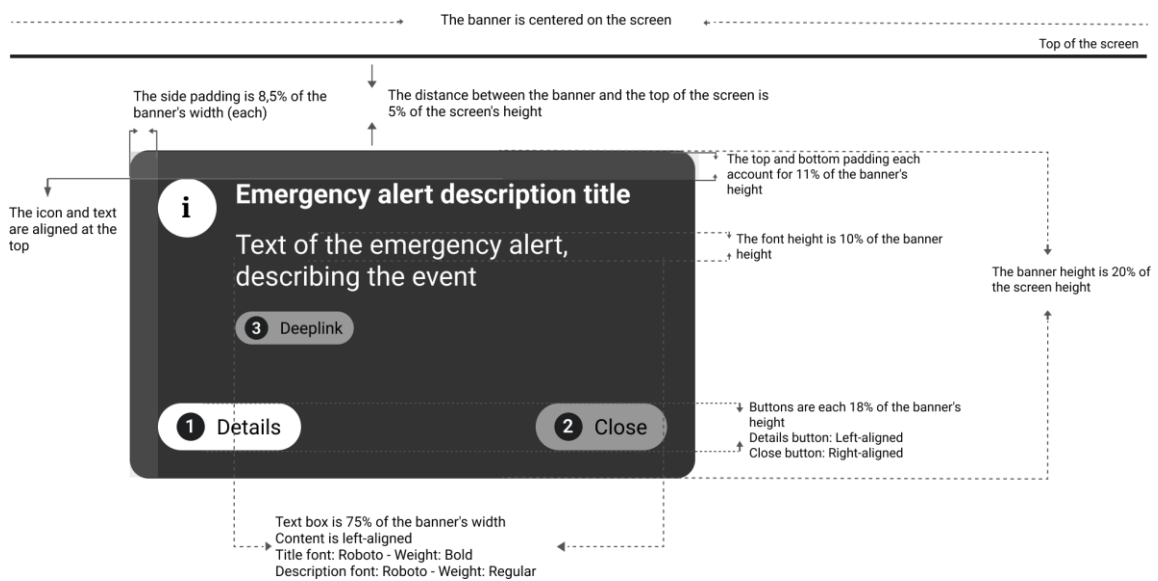


Figure 10 – Recommended positioning of the alert window when a deep link button is displayed.

An example of the moderate alert on TV is shown in Figure 11.

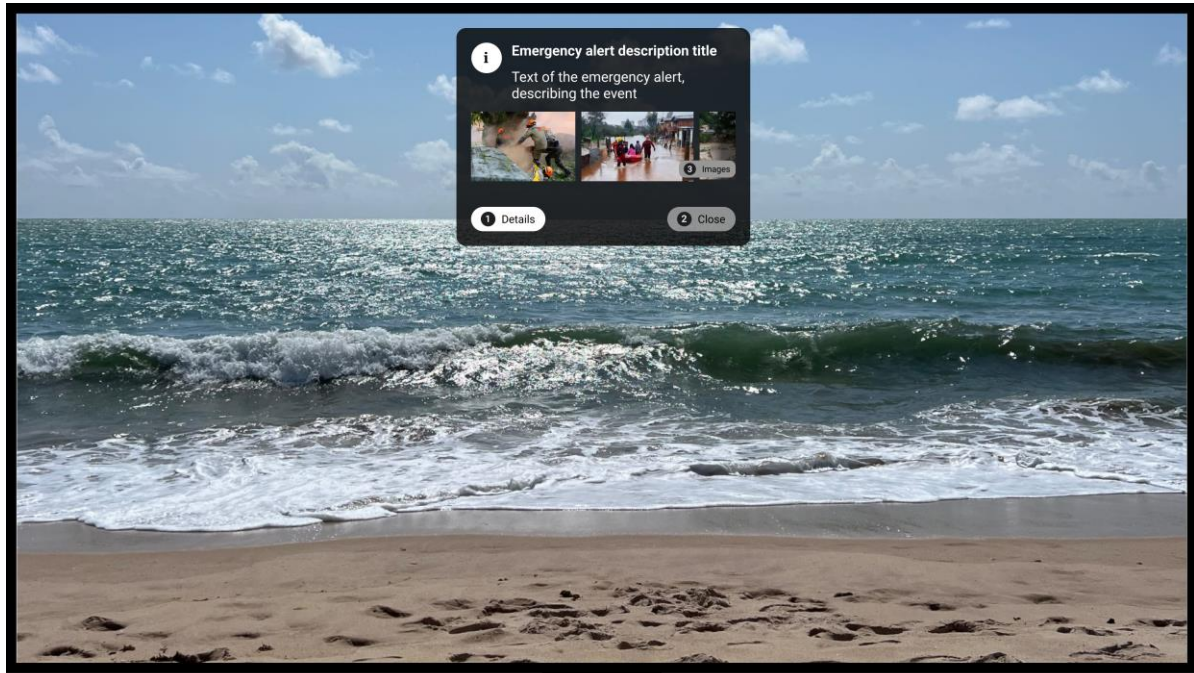


Figure 11 – Illustration of the moderate priority alert display on a 16:9 TV.

9.2.4. High Priority

The display recommendations for high-priority alerts are shown in Figures 12 to 16. Figure 12 shows the proposed standard layout, with recommendations for positioning, font, aspect ratio, and the options to 'Details' and 'Close'.

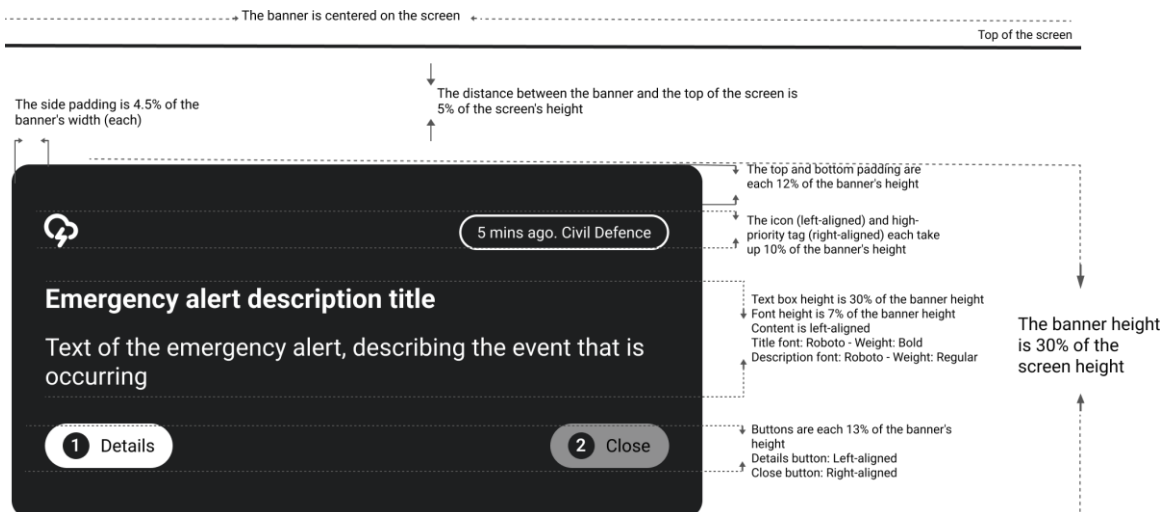


Figure 12 – Recommendations for the positioning of the alert window, font and screen aspect ratio.

Figure 13 shows the expanded layout after clicking the action button (Details).

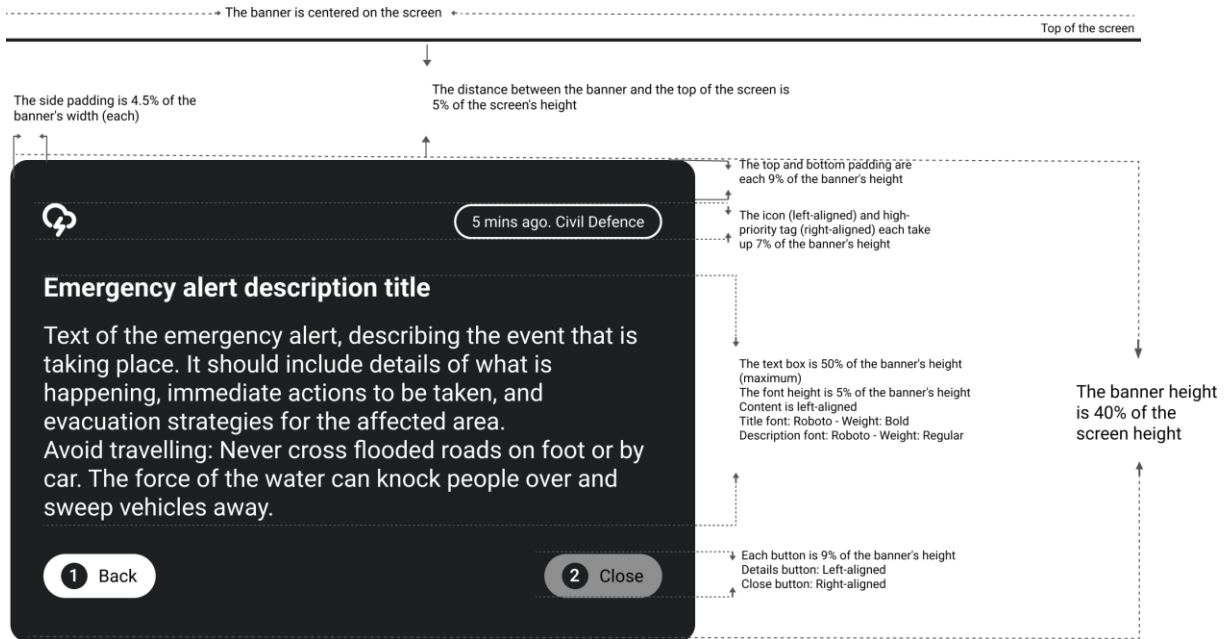


Figure 13 – Recommendations for the positioning of the expanded alert window, font and screen aspect ratio.

Figure 14 and Figure 15 show the recommended display of the sign language window for alerts transmitted with accessibility features.

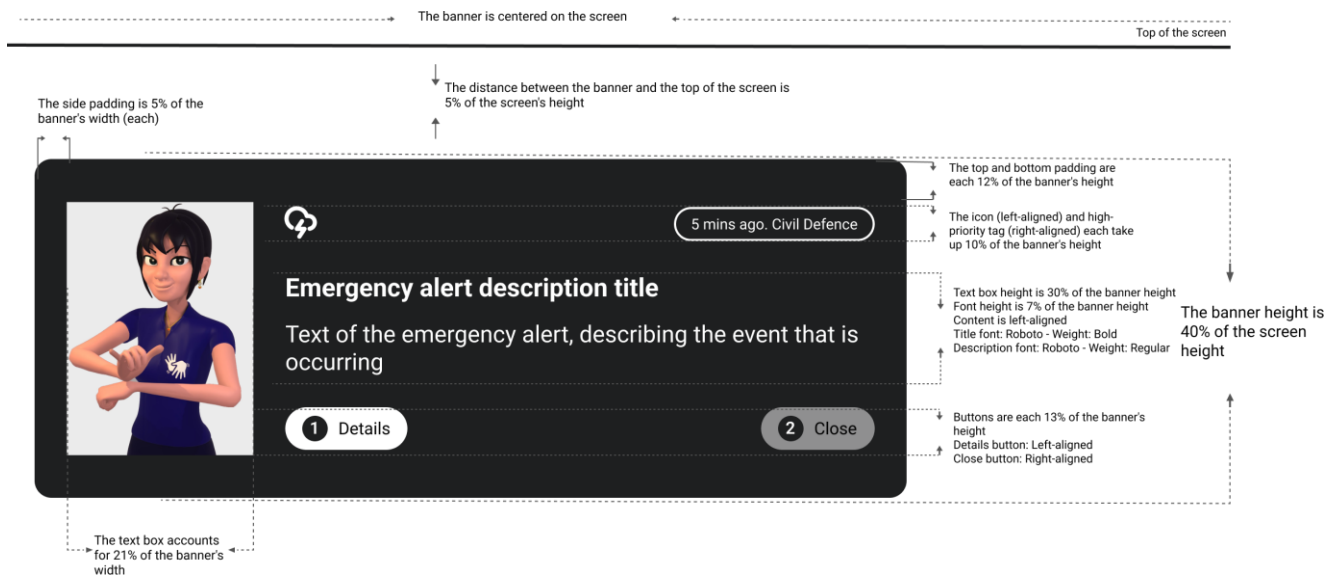


Figure 14 – Recommended positioning of the alert window, font and aspect ratio for the screen displaying the sign language window.

Figure 15 shows the expanded layout, with the sign language window, after clicking the action button (Details).

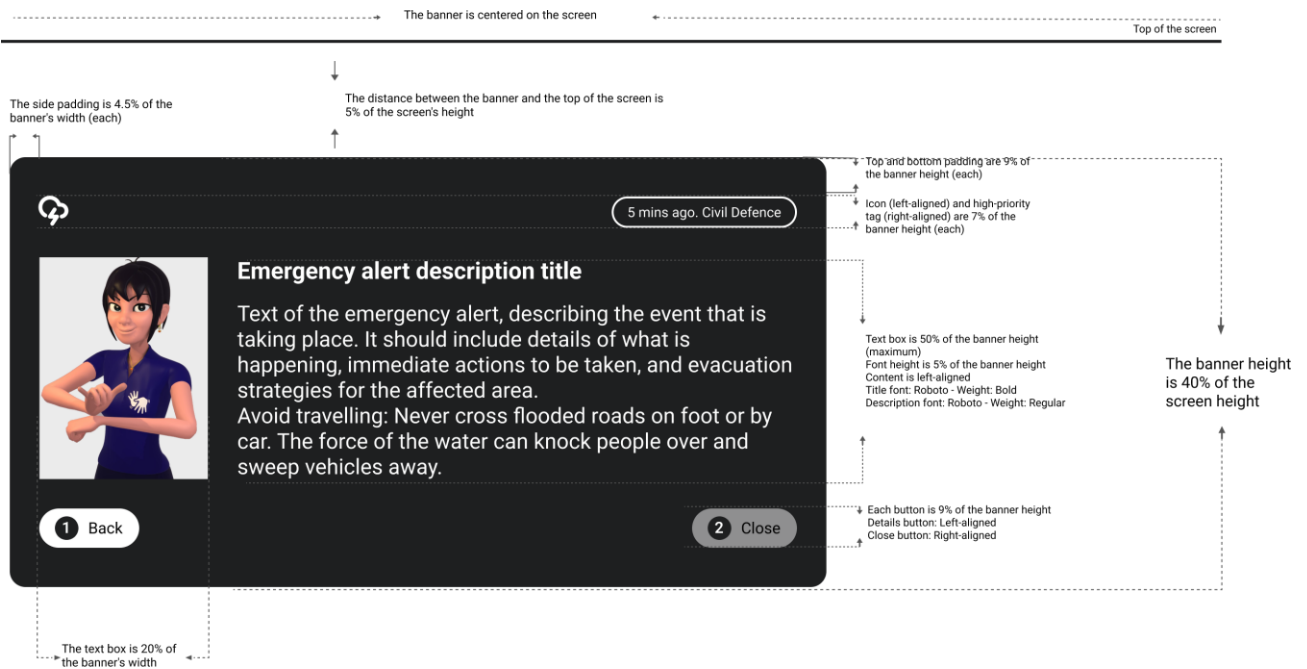


Figure 15 – Recommended positioning of the expanded alert window, font and aspect ratio for the screen displaying the sign language window.

An example of the high priority alert on TV is shown in Figure 16.

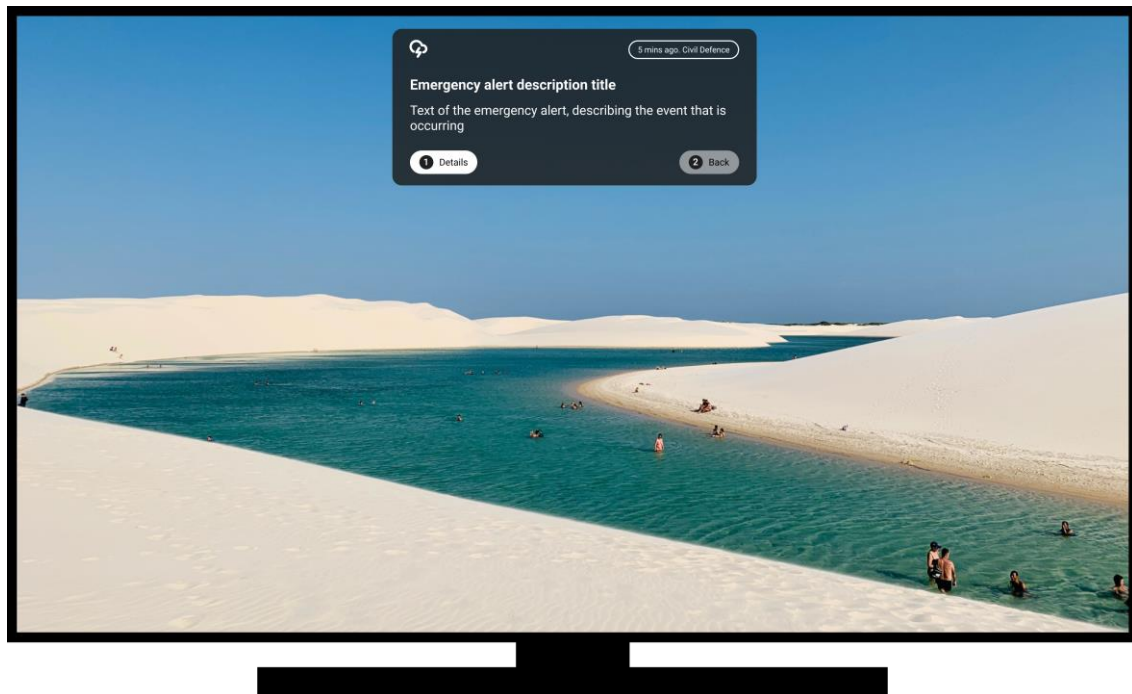


Figure 16 – Illustration of the high priority alert display on a 16:9 TV.

9.2.5. Maximum Priority

The display recommendations for high-priority alerts are shown in the following figures. In all proposals, the alert is displayed in full-screen mode. Figure 17 shows an example of an alert containing only textual information.

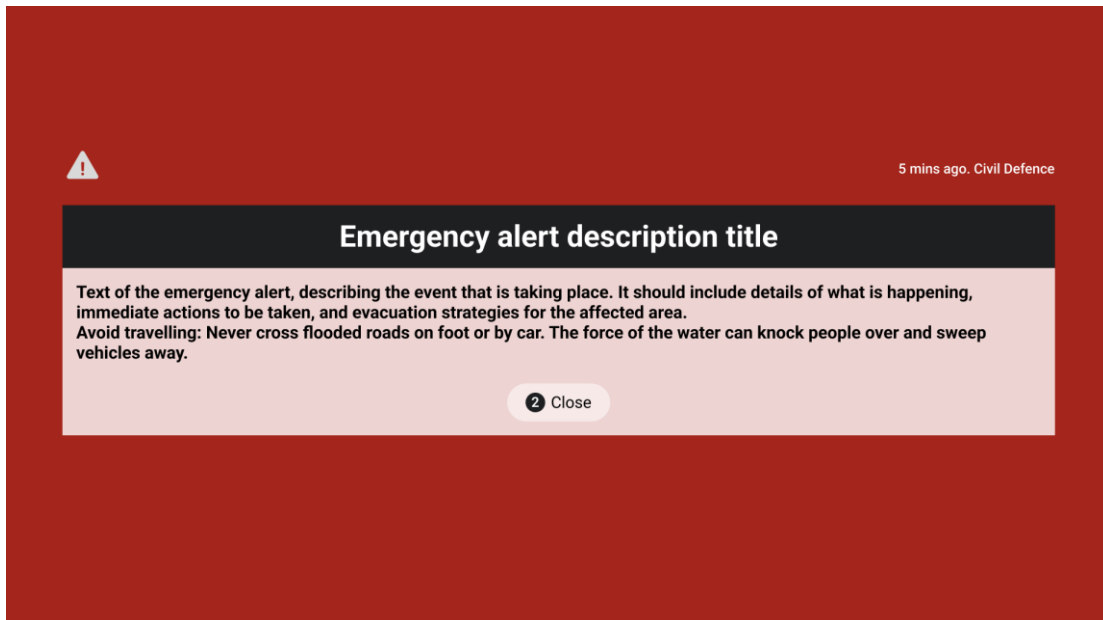


Figure 17 – Illustration of a high-priority alert display containing only textual information.

Figure 18 shows an alert with text and multiple images.

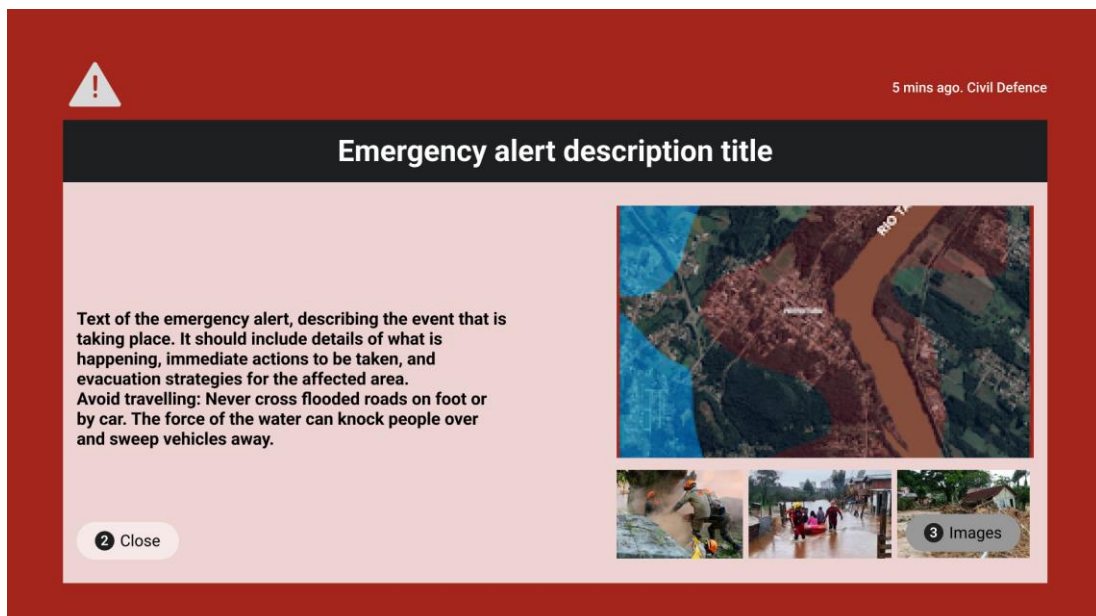


Figure 18 – Illustration of the priority alert display showing text and multiple images.

Figure 19 shows the recommended font and aspect ratio for the screen displaying an image and the sign language window and images.

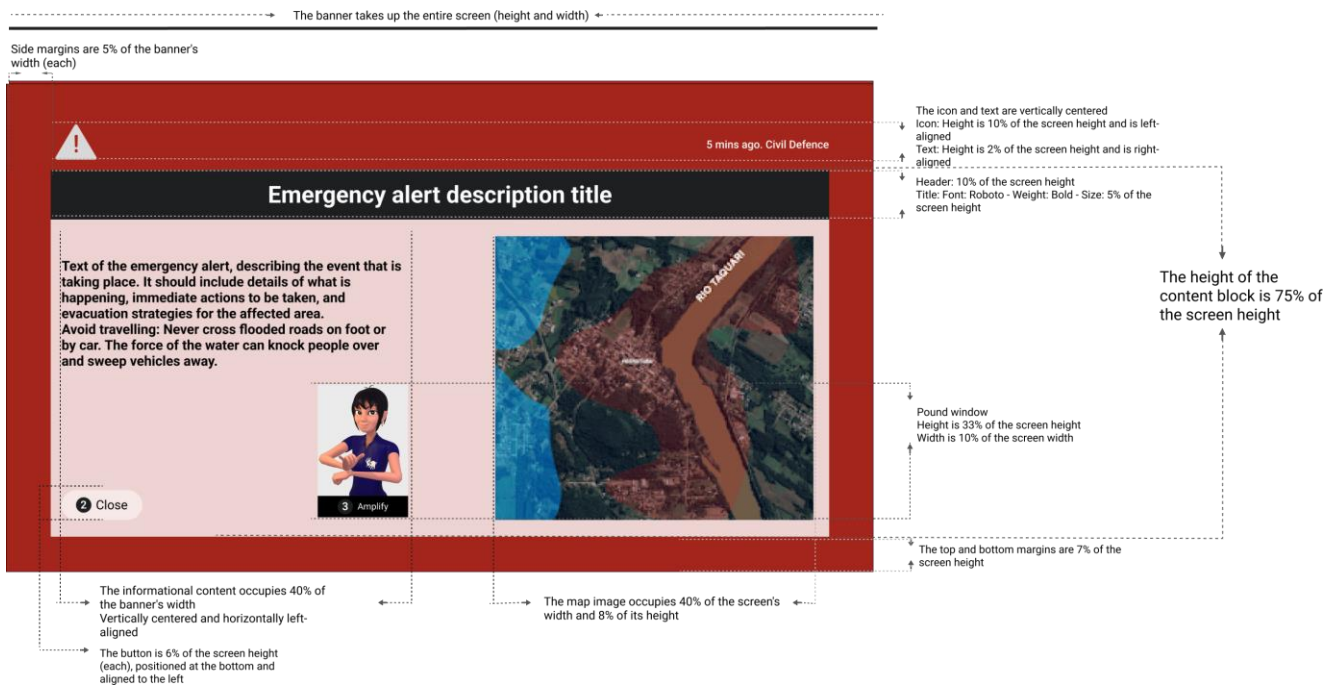


Figure 19 – Recommended positioning of the alert window, font and aspect ratio for the screen displaying an image and the sign language window and images.

Figure 20 shows how a deep link to applications can be made available via a button.

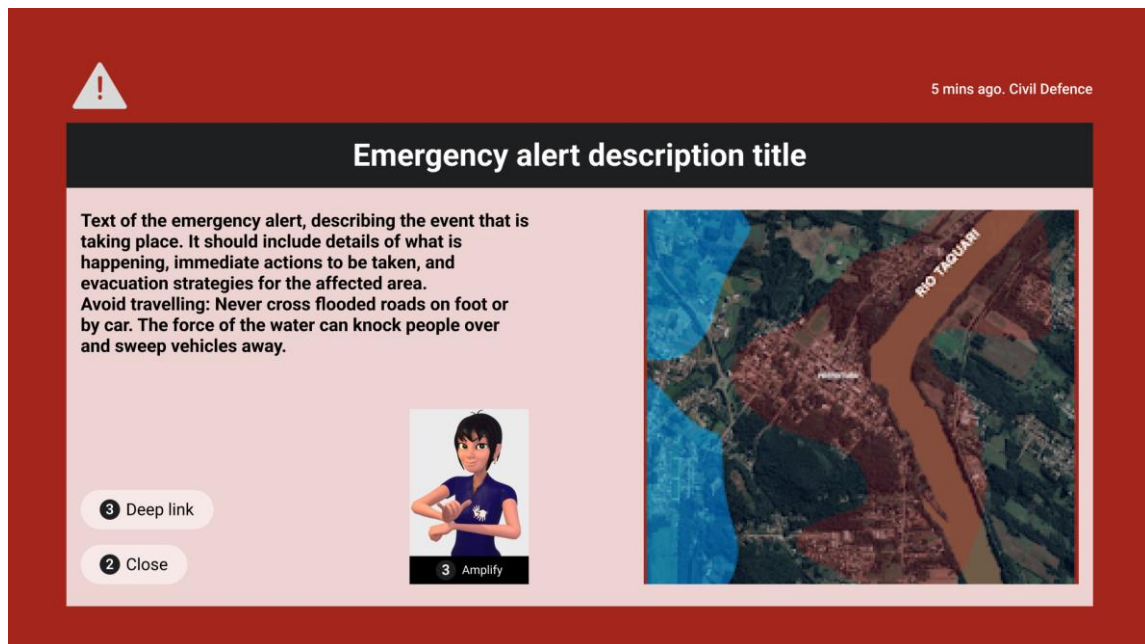


Figure 20 – Recommended positioning of the alert window when a deep link button is displayed.

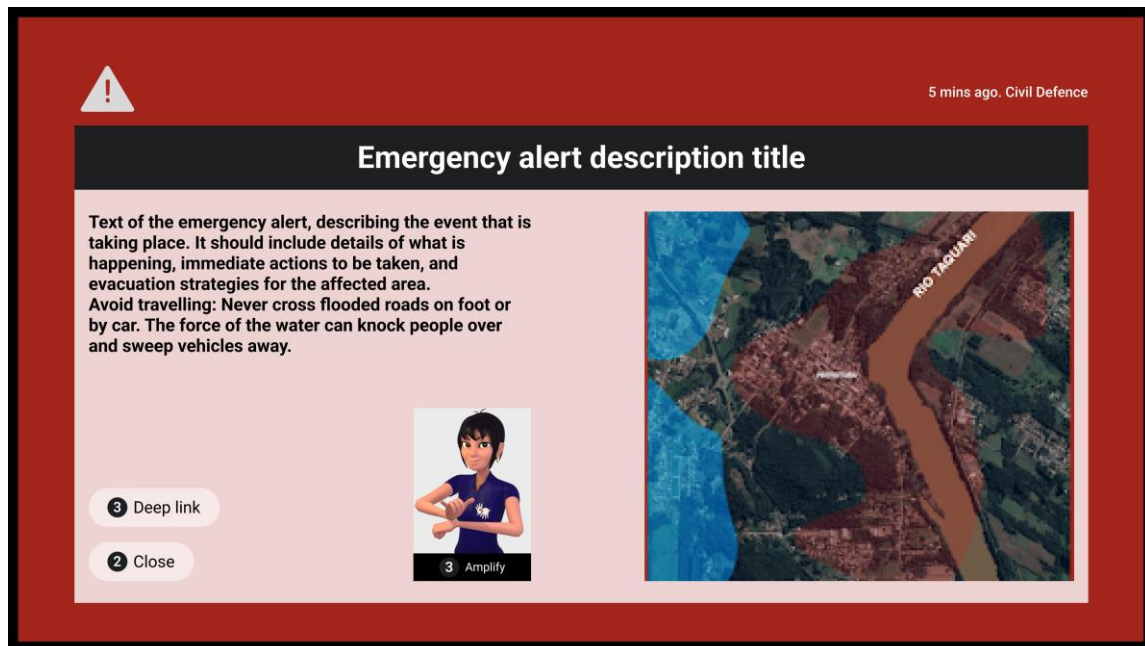


Figure 21 – Illustration of the maximum priority alert display on a 16:9 TV.

Annex A

Regulatory Compliance Requirements for Brazil

This Annex details and explains recommendations and behavior for AEAT configuration and format and the Broadcaster Station Operation for Brazil.

A.1 AEAT configuration and format in Brazil

Several AEAT fields should be configured according to the descriptions found in Table A.1 when used at Brazil.

Table A.1 - Mapping of the fields and configurations.

Field	Configuration
<code>AEA.Header.EventCode</code>	This string element should identify the event type of the AEA message formatted as a string. Values may differ from nation to nation, and may be an alphanumeric code, or may be plain text. For Brazil, the values should be a code number formatted with 6 numbers, from the Brazilian Disaster Classification and Coding (COBRADE) [1]. E.g., a value of "1.2.2.0.0" would be used to denote flooding (<i>enxurradas</i>).
<code>AEA.Header.EventCode@type</code>	This attribute should be a national-assigned string value that should designate the domain of the EventCode. Values of @type that are acronyms should be represented in all capital letters without periods. If @type="COBRADE", then the EventCode should be defined as code number formatted with 5 numbers, from the COBRADE [1].
<code>AEA.Header.Location@type</code>	A string that should identify the domain of the Location code. The following values have specific behaviors on Brazil: <ul style="list-style-type: none"> "polygon". Alerts distributed through the Public Alert Dissemination Interface (IDAP) always contain polygon-type locations and are transmitted in CAP format. "ZIP_BR". When <code>AEA.Header.Location@type</code> is "ZIP_BR", <code>AEA.Header.Location</code> must represent a ZIP code for a specific location. Post office box ZIP codes and promotional ZIP codes are not valid.

A.2 Broadcaster Station Operation

Broadcasters transmitting emergency alerts via OTT are advised to monitor the Emergency Alert Authorities Web Server. For Brazil, it is recommended to monitor the Public Alert Notification Interface (IDAP) (<https://idapfile.mi.gov.br/idap/api/rss/cap>). This service provides alerts for the last 24 hours.

Annex B

Conversion script from CAP to AEAT

An example of conversion script has been made available, and it's presented on Table B.1. It takes an XML file in CAP format as input and generates an XML file in AEAT format as output, containing the conversion results.

Table B.1 - Conversion script from CAP to AEAT

```
#!/usr/bin/env python3
"""
cap_to_aeat.py - CAP (Common Alerting Protocol) to AEAT (AEA Table) converter
Usage:
    python cap_to_aeat.py <input.xml> <output.xml>
    python cap_to_aeat.py <input.xml>          # output: <input>_aeat.xml
Based on the conversion table from the Emergency Warning System Operational Guide v1.
"""
import sys
import uuid
from pathlib import Path
from lxml import etree
# CAP namespace
CAP_NS = "urn:oasis:names:tc:emergency:cap:1.2"

def _cap(tag: str) -> str:
    return f"{{{CAP_NS}}}{tag}"
# CAP reading helpers
def text(el, *path):
    """Traverse a chain of child tags and return the text of the last element."""
    cur = el
    for tag in path:
        cur = cur.find(_cap(tag)) if cur is not None else None
    return cur.text.strip() if cur is not None and cur.text else None

def findall(el, *path):
    """Return all child elements at the last level of the path."""
    cur = el
    for tag in path[:-1]:
        cur = cur.find(_cap(tag)) if cur is not None else None
    if cur is None:
        return []
    return cur.findall(_cap(path[-1]))
# Priority mapping (Table 2)
def map_priority(severity: str | None, urgency: str | None) -> str:
    if severity == "Extreme" and urgency == "Immediate":
        return "4" # Maximum
    if severity == "Severe" and urgency == "Immediate":
        return "3" # High
    if severity == "Moderate":
        return "2" # Moderate
    if severity == "Minor":
        return "1" # Low
    return "0" # Minor / Unknown
# Area -> Location mapping
def add_location(header_el, value: str, loc_type: str):
    loc = etree.SubElement(header_el, "Location")
    loc.set("type", loc_type)
    loc.text = value
# Main conversion
def convert(cap_path: str, aeat_path: str) -> None:
    tree = etree.parse(cap_path)
    root = tree.getroot()
    # Support explicit namespace or no namespace
    alert = root if root.tag in (_cap("alert"), "alert") else root.find(_cap("alert"))
    if alert is None:
        alert = root

def t(*path):
```

```

return text(alert, *path)

def fa(*path):
    return findall(alert, *path)
# AEAT root element
aeat_table = etree.Element("AEAT")
aeat_table.set("aeaTableId", str(uuid.uuid4())) # no CAP equivalent → generated UUID
# AEA element
aea = etree.SubElement(aeat_table, "AEA")
# @aeaId ← cap.alert.identifier
identifier = t("identifier")
if identifier:
    aea.set("aeaId", identifier)
# @sender ← cap.alert.sender
sender = t("sender")
if sender:
    aea.set("sender", sender)
# @issuer ← cap.alert.source (N/A when absent)
source = t("source")
if source:
    aea.set("issuer", source)
# @aeaType ← cap.alert.msgType
msg_type = t("msgType")
if msg_type:
    aea.set("aeaType", msg_type)
# @audience ← cap.alert.scope
scope = t("scope")
if scope:
    aea.set("audience", scope)
# @subAudience ← concat(restriction, addresses, audience info)
sub_audience_parts = []
restriction = t("restriction")
if restriction:
    sub_audience_parts.append(restriction)
addresses = t("addresses")
if addresses:
    sub_audience_parts.append(addresses)
# @refAEAId ← cap.alert.references
references = t("references")
if references:
    aea.set("refAEAId", references)
# @wakeUp – no CAP equivalent; defaults to false (may be overridden by priority rule)
aea.set("wakeUp", "false")
infos = fa("info")
if not infos:
    infos = []
for info in infos:
    def i_text(*path):
        return text(info, *path)
    # Priority derived from severity + urgency
    severity = i_text("severity")
    urgency = i_text("urgency")
    priority = map_priority(severity, urgency)
    aea.set("priority", priority)
    if priority == "4": # Maximum → wakeUp = true
        aea.set("wakeUp", "true")
    # @category ← cap.alertInfo.category
    category = i_text("category")
    if category:
        aea.set("category", category)
    # Append info audience to @subAudience concat parts
    audience_info = i_text("audience")
    if audience_info:
        sub_audience_parts.append(audience_info)
    # Write @subAudience after collecting all parts
    if sub_audience_parts:
        aea.set("subAudience", " ".join(sub_audience_parts))
    lang = i_text("language")
    # Header element
    header = etree.SubElement(aea, "Header")
    # @effective: priority 1=effective, 2=onset, 3=sent

```

```

effective = i text("effective") or i text("onset") or t("sent")
if effective:
    header.set("effective", effective)
# @expires ← cap.alertInfo.expires
expires = i_text("expires")
if expires:
    header.set("expires", expires)
# EventDesc ← cap.alertInfo.event
event = i text("event")
if event:
    event_desc = etree.SubElement(header, "EventDesc")
    if lang:
        event_desc.set("lang", lang)
    event_desc.text = event
# EventCode(s) ← cap.alertInfo.eventCode
for ev_code in info.findall(_cap("eventCode")):
    value_name = text(ev_code, "valueName")
    value = text(ev_code, "value")
    ec_el = etree.SubElement(header, "EventCode")
    if value_name:
        ec_el.set("type", value_name)
    if value:
        ec_el.text = value
# Location ← polygon, circle, geocode
for polygon in info.findall( cap("area") + "/" + cap("polygon")):
    if polygon.text:
        add_location(header, polygon.text.strip(), "polygon")
for circle in info.findall( cap("area") + "/" + cap("circle")):
    if circle.text:
        add_location(header, circle.text.strip(), "circle")
for geocode in info.findall(_cap("area") + "/" + _cap("geocode")):
    val_name = text(geocode, "valueName")
    val = text(geocode, "value")
    if val:
        loc_type = val_name if val_name else "geocode"
        add_location(header, val, loc_type)
# AEAText ← concat(headline, description, instruction)
text_parts = []
headline = i_text("headline")
description = i text("description")
instruction = i text("instruction")
if headline:
    text_parts.append(headline)
if description:
    text_parts.append(description)
if instruction:
    text_parts.append(instruction)
if text_parts:
    aeatext = etree.SubElement(aea, "AEAText")
    if lang:
        aeatext.set("lang", lang)
    aeatext.text = " ".join(text_parts)
# Media ← cap.alertInfo.resource
for resource in info.findall(_cap("resource")):
    media = etree.SubElement(aea, "Media")
    if lang:
        media.set("lang", lang)
    res_desc = text(resource, "resourceDesc")
    if res_desc:
        media.set("mediaDesc", res_desc)
    mime = text(resource, "mimeType")
    if mime:
        media.set("contentType", mime)
    uri = text(resource, "uri")
    if uri:
        media.set("url", uri)
    size = text(resource, "size")
    if size:
        media.set("contentLength", size)
    # derefUri and digest have no AEAT equivalent – ignored
# Serialization

```

```
out tree = etree.ElementTree(aeat table)
out tree.write(aeat path, xml declaration=True, encoding="UTF-8", pretty print=True)
print(f"✓ AEAT file generated: {aeat_path}")
# Entry point
def main():
    if len(sys.argv) < 2:
        print("Usage: python cap_to_aeat.py <input.xml> [output.xml]")
        sys.exit(1)
    cap_file = sys.argv[1]
    if len(sys.argv) >= 3:
        aeat_file = sys.argv[2]
    else:
        stem = Path(cap_file).stem
        aeat_file = str(Path(cap_file).parent / f"{stem}_aeat.xml")
    convert(cap_file, aeat_file)

if name == " main ":
    main()
```

Bibliography

- [1] COBRADE. Classificação Brasileira de Desastres 2017. Disponível em: <https://educacao.cemaden.gov.br/midiateca/classificacao-e-codificacao-brasileira-de-desastres-cobrade/>. Acesso em: 08/06/2026.